stats_ch02_what_are_data

July 27, 2024

1 Modern statistics: Intuition, Math, Python, R

1.1 Mike X Cohen (sincxpress.com)

https://www.amazon.com/dp/B0CQRGWGLY

Code for chapter 2 (What are data?)

---

2 About this code file:

2.0.1 This notebook will reproduce most of the figures in this chapter (some figures were made in Inkscape), and illustrate the statistical concepts explained in the text. The point of providing the code is not just for you to recreate the figures, but for you to modify, adapt, explore, and experiment with the code.

2.0.2 Solutions to all exercises are at the bottom of the notebook.

This code was written in google-colab. The notebook may require some modifications if you use a different IDE.

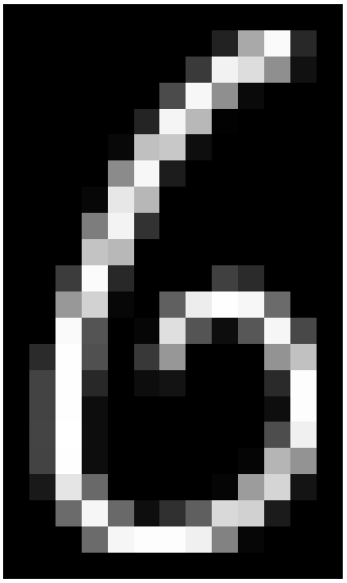
[1]:
```
# import libraries and define global settings
import numpy as np
import matplotlib.pyplot as plt

# define global figure properties used for publication
#import matplotlib_inline.backend_inline
#matplotlib_inline.backend_inline.set_matplotlib_formats('svg') # display␣
 ↪figures in vector format
#plt.rcParams.update({'font.size':14,             # font size
#                     'savefig.dpi':300,          # output resolution
#                     'axes.titlelocation':'left',# title location
#                     'axes.spines.right':False,  # remove axis bounding box
#                     'axes.spines.top':False,    # remove axis bounding box
#                     })
```

[ ]:
```

```

3 Figure 2.3: The "6" data, as an image and numbers

[2]:
```
# import MNIST data
from sklearn.datasets import fetch_openml
mnist = fetch_openml('mnist_784', as_frame=False, cache=False, parser='auto')
```

[3]:
```
# show one number
_,axs = plt.subplots(1,2,figsize=(8,6))

# the image of the number
I = mnist.data[18].reshape(28,28)[2:24,:][:,8:-7]
axs[0].imshow(I,cmap='gray')
axs[0].axis('off')

axs[1].imshow(I,cmap='gray',vmin=-1,vmax=0)
axs[1].axis('off')

# and the numbers of the number
for i in range(I.shape[0]):
  for j in range(I.shape[1]):
    axs[1].text(j,i,int(I[i][j]),fontsize=8
                ,horizontalalignment='center',verticalalignment='center')

plt.tight_layout()
#plt.savefig('whatR_mnist.png')
plt.show()
```

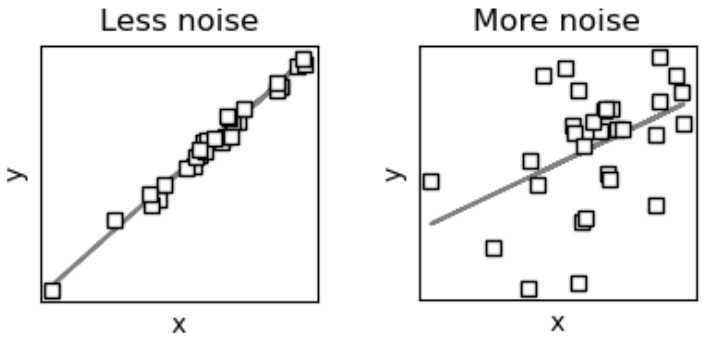


```
  0   0   0   0   0   0   0   0   0   0   0   0   0
  0   0   0   0   0   0   0   0   0  34 169 250  40   0
  0   0   0   0   0   0   0   0  58 242 221 143  17   0
  0   0   0   0   0   0   0  75 247 143  10   0   0   0
  0   0   0   0   0   0  37 245 184   2   0   0   0   0
  0   0   0   0   8 192 200  14   0   0   0   0   0   0
  0   0   0   0 139 247  28   0   0   0   0   0   0   0
  0   0   0   7 231 183   0   0   0   0   0   0   0   0
  0   0   0 125 243  50   0   0   0   0   0   0   0   0
  0   0   0 195 184   0   0   0   0   0   0   0   0   0
  0   0  61 251  41   0   0   0  64  43   0   0   0   0
  0   0 152 210   7   0  96 237 254 247 107   0   0   0
  0   0 250  84   0   6 223  84  13  87 246  72   0   0
  0  43 254  80   0  56 151   0   0   0 147 193   0   0
  0  67 254  41   0  13  19   0   0   0  42 253   0   0
  0  67 254  13   0   0   0   0   0   0  14 253   0   0
  0  68 255  13   0   0   0   0   0   0  77 240   0   0
  0  67 254  13   0   0   0   0   0   5 181 147   0   0
  0  25 229 105   0   0   0   0   5 156 213  20   0   0
  0   0 107 246 105  14  49  95 217 209  27   0   0   0
  0   0   0 107 246 253 253 240 130   6   0   0   0   0
  0   0   0   0   0   0   0   0   0   0   0   0   0   0
```

[ ]:
```

```

4 Figure 2.5: Margin figure with noisy data

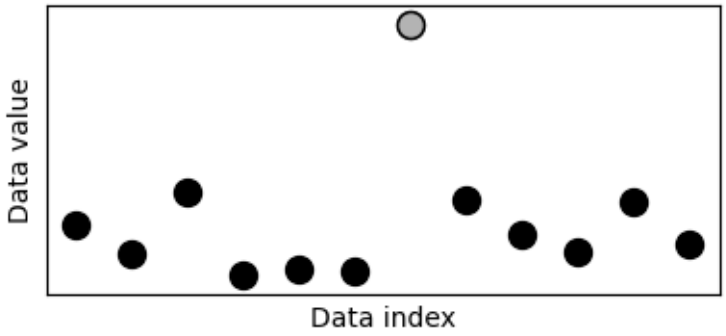
[4]:
```
# generate data
n = 30
x = np.random.randn(n)
y1 = x + np.random.randn(n)/10
y2 = x + np.random.randn(n)

_,axs = plt.subplots(2,1,figsize=(2,4))

axs[0].plot(x,np.polyval(np.polyfit(x,y1,1),x),color='gray')
axs[0].plot(x,y1,'ws',markeredgecolor='k')
axs[0].set_title('Less noise',loc='center')

axs[1].plot(x,np.polyval(np.polyfit(x,y2,1),x),color='gray')
axs[1].plot(x,y2,'ws',markeredgecolor='k')
axs[1].set_title('More noise',loc='center')

for a in axs:
  a.set_xticks([])
  a.set_xlabel('x')
  a.set_yticks([])
  a.set_ylabel('y')

plt.tight_layout()
#plt.savefig('whatR_noisyData.png')
plt.show()
```



[ ]:
```

```

5 Figure 2.6: Margin figure with outlier

[5]:
```
X = np.random.randn(12)
X[6] = 2*np.pi

plt.figure(figsize=(4,2))
plt.plot(X,'ko',markersize=10)
plt.plot(6,X[6],'ko',markersize=10,markerfacecolor=(.7,.7,.7))
plt.xticks([])
plt.yticks([])
plt.ylim([np.min(X)-.6,np.max(X)+.6])
plt.xlabel('Data index')
plt.ylabel('Data value')

plt.tight_layout()
#plt.savefig('whatR_outlier.png')
plt.show()
```



[ ]:
```

```